

owing to the process of acquiring a new IP address at the time of handover, although the delay due to the process by NTMobile is quite small.

## IPv4/IPv6 混在環境で移動透過性を実現する NTMobile の実装と評価

上 醉尾 一真<sup>†1</sup>    鈴木 秀和    †1  
内 藤 克 浩    †2    渡 邊 晃    †1

現在の IP ネットワークは IPv4 から IPv6 への過渡期にあり、今後は互換性のない IPv4 と IPv6 が混在したネットワークになることが想定される。我々は、IPv4 と IPv6 が混在した環境において確実な接続性の確保と、通信中のネットワーク切り替えを可能とする移動透過性を同時に実現する NTMobile (Network Traversal with Mobility) を提案している。本稿では NTMobile を Android OS を搭載したスマートフォンへ実装し、ハンドオーバーなどの動作検証および性能評価を行った。動作検証の結果、IPv4 ネットワークと IPv6 ネットワーク間における相互接続性の確保、および移動透過性を実現可能であることを確認した。また、NTMobile による処理遅延はわずかであるものの、ハンドオーバー時には IP アドレス取得処理に起因する通信切断時間が発生することがわかった。

### Implementation and Evaluation of NTMobile that realize IP Mobility in IPv4 and IPv6 Networks

KAZUMA KAMIENOO,<sup>†1</sup> HIDEKAZU SUZUKI,<sup>†1</sup>  
KATSUHIRO NAITO<sup>†2</sup> and AKIRA WATANABE<sup>†1</sup>

As the IP network system is currently in a transitional period from IPv4 to IPv6, it is thought that we have an IPv4 and IPv6 networks hereafter in which both systems do not have compatibility. We have been proposing a system which we named “NTMobile” (Network Traversal with Mobility) that realizes both connectivity and mobility at the same time in IPv4 and IPv6 networks. In this paper, we show the results of implementation of NTMobile in a smartphone equipped with an Android OS, together with the results of verification of its operability such as handover and our evaluation of its performance. As the result of our verification, it was confirmed that it is possible to make mutual connectivity between the IPv4 network and the IPv6 network as well as mobility. It was also found that there is a period of communication interruption

#### 1. はじめに

現在の IP ネットワークは IPv4 から IPv6 への過渡期にあり、IPv4 と IPv6 が共存した環境が広まりつつある。しかし、IPv4 と IPv6 には互換性がないため、これらのネットワーク間において相互に通信を行うことができない。一方、既存の IPv4 ネットワークでは NAT を導入してプライベートネットワークを構築することが一般的であり、LSN (Large Scale NAT) のようにキャリアレベルでも NAT が導入され始めている。NAT が導入された環境においては、グローバルネットワーク側の端末からプライベートネットワーク側の端末に対する接続性を確保できない、NAT 越え問題と呼ぶ課題があり、エンドツーエンドの接続性というインターネット本来の理念を損なう要因となっている。これまでに NAT 越えを実現する技術が提案されてきたが、冗長経路問題やシグナリングオーバーヘッド、NAT やプライマリ DNS サーバ等の特定の装置に改造が必要になるなどの課題がある [1-3]。今後の IP ネットワークを想定すると、NAT 環境や IPv4 と IPv6 が混在した環境においても、確実に接続性を確保する技術が必要である。

また、スマートフォンなどの高性能な携帯端末の普及や無線技術の発展により、移動しながら通信を行いたいという要求が高まっている。スマートフォンには 3G や Wi-Fi, WiMAX など複数の無線インタフェースが搭載されており、必要に応じてインタフェースを切り替えて通信を行うことができる。しかし、IP ネットワークでは通信端末のインタフェースに割り当てられた IP アドレスを用いて通信を管理しているため、インタフェースやネットワークの切り替えに伴い IP アドレスが変化すると通信を継続することができない。このような問題を解決する技術を移動透過性技術と呼び、現在までに様々な移動透過性技術が提案されている [4-10]。既存の移動透過性技術の多くは IPv6 ネットワークを想定しており、IPv4 ネットワークへの適用が検討されている場合であっても、NAT 環境において移動や通信が制限されることや、経路が冗長になるという課題がある。

---

†1 名城大学大学院理工学研究科

Graduate School of Science and Technology, Meijo University

†2 三重大学大学院工学研究科

Graduate School of Engineering, Mie University

IPv4/IPv6 混在環境において移動透過性と NAT 越えを実現する技術として、DSMIPv6 (Dual Stack Mobile IPv6) が提案されている。DSMIPv6 では、ホームネットワークに設置した HA (Home Agent) が通信を中継することにより、IPv4/IPv6 ネットワーク間の通信や NAT 越えを実現している。しかし、DSMIPv6 では、IPv4 ネットワークにおいて常に HA を経由した冗長経路となることや、HA の一点障害が課題となっている。

著者らは、IPv4 と IPv6 が混在した環境において確実な接続性と移動透過性を同時に実現する NTMobile (Network Traversal with Mobility) を提案している [11-14]。NTMobile は NAT 越えの技術を兼ね備えており、NAT に改造を加えることなく NAT 配下の移動端末 (以後 NTM 端末) に対する接続性を確保することができる。NTMobile では NTM 端末に仮想 IP アドレスを割り当て、アプリケーションが仮想 IP アドレスに基づいた通信を行うことにより、端末の移動に伴う実 IP アドレスの変化を隠蔽し、アプリケーション間の通信を継続する。また、NAT の有無や IPv4 ネットワークと IPv6 ネットワーク間の通信など、通信端末が接続しているネットワークに応じて最適な経路でトンネルを構築し、アプリケーションが生成したパケットを転送する。NTM 端末間に構築されるトンネルは、特定の状況を除きエンドツーエンドで構築されるため、経路が冗長になりにくい。本稿では NTMobile を Android OS<sup>\*1</sup> を搭載したスマートフォンへ実装し、IPv4 と IPv6 が混在した環境における動作検証および性能評価を行った。

以下、2 章で関連研究の課題について述べ、3 章で NTMobile について概説する。4 章で実装および性能評価について述べ、5 章でまとめる。

## 2. 関連研究

本章では、IPv4 と IPv6 の混在環境において移動透過性を実現する、DSMIPv6 (Dual Stack Mobile IPv6) [15] の概要と課題について述べる。DSMIPv6 は、IPv6 ネットワークにおいて移動透過性を実現する Mobile IPv6 [6] を IPv4/IPv6 混在環境へ適用するために拡張した技術である。DSMIPv6 では移動端末に対して、ホームネットワークで取得する HoA (Home Address) と訪問先ネットワークから取得する CoA (Care of Address) の二種類のアドレスを割り当て、アプリケーションが HoA を用いた通信を行うことにより、端末の移動に伴う CoA の変化を隠蔽する。

図 1 に DSMIPv6 の概要を示す。移動端末 MN はホームネットワークに設置した HA

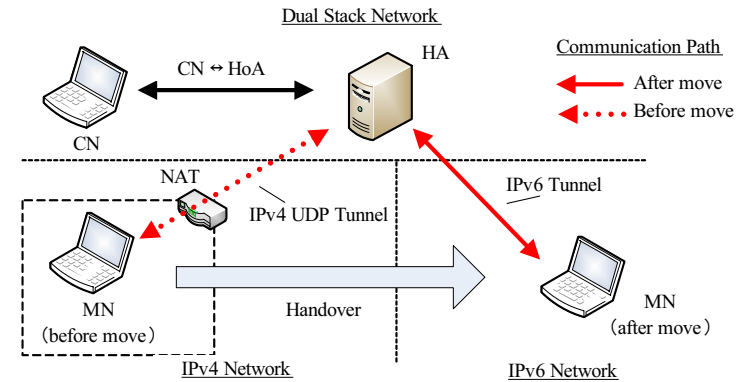


図 1 DSMIPv6 の概要  
Fig.1 Overview of DSMIPv6

(Home Agent) との間にトンネルを構築し、HA を経由して通信相手 CN と通信を行う。ホームネットワークはデュアルスタックネットワークとして構築されており、HA は IPv4 ネットワークと IPv6 ネットワーク間の橋渡しを行う。MN のアプリケーションが生成したパケットはトンネルを用いて HA へ送信され、HA によりデカプセル化されたあと、CN へ送信される。これにより、CN は通信相手の IP アドレスとして HoA を認識することになり、MN のアプリケーションと CN 間には HoA に基づいたコネクションが確立される。HoA は移動により変化しないため、通信中に MN が移動をしても、アプリケーションや CN に対して移動を隠蔽し、通信を継続することができる。また、IPv4 の HoA を用いることにより、IPv6 ネットワーク上にて IPv4 アプリケーションを使用することができる。

Mobile IPv6 には、経路最適化を行うことにより MN と CN 間にてエンドツーエンドの通信を行う機能が定義されていた。しかし、この機能は IPv6 特有のヘッダオプションを使用するため、DSMIPv6 において通信端末が IPv4 ネットワークに位置する場合や IPv4 アプリケーションを使用している場合には適用することができない。また、DSMIPv6 では MN が NAT 配下に接続している場合、常に HA を介した冗長な経路で通信を行うか、訪問先のネットワークに特殊な NAT を設置する必要がある [16]。

また、Mobile IPv6 では HA への負荷分散や経路の冗長性を抑制するために、Global HA to HA Protocol と呼ぶ技術が議論されている [17]。この技術では、複数の HA をホーム

\*1 Android OS とは、米 Google 社が発表した Linux をベースとした携帯端末向けの OS である。

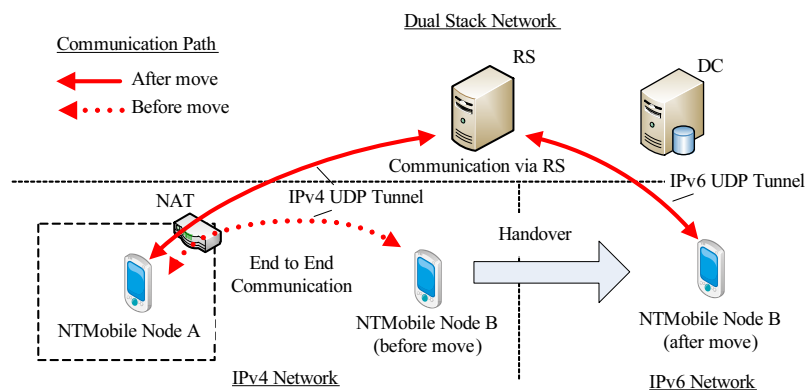


図 2 NTMobile の概要  
Fig. 2 Overview of NTMobile.

ネットワーク外に分散設置し、HA 同士がオーバーレイネットワークを構築する。MN の通信を中継する際には、経路的に近い HA が中継装置として選ばれる。これにより、HA の一点障害の解決や、経路の冗長性を抑制することができる。しかし、この機能は DSMIPv6 や IPv4 ネットワークへの適用は議論されていないため、DSMIPv6 によるシステムでは HA の一点障害や冗長経路が発生するという課題がある。

### 3. NTMobile

#### 3.1 概要

NTMobile では、実際のネットワークに依存しない仮想 IP アドレスを導入することにより、IPv4 ネットワークと IPv6 ネットワーク上の端末を統一的に管理する。NTM 端末のアプリケーションは仮想 IP アドレスに基づいたコネクションを確立することにより、ネットワークの切り替えや接続しているネットワークの違いに影響されることなく、自由に通信を行う事ができる。仮想 IP アドレスに基づくパケットは、NTM 端末間に構築される UDP トンネルによって送信される。このトンネルは特定の状況を除きエンドツーエンドで構築されるため、通信端末は常に最適な経路で通信を行うことができる。

NTMobile のネットワーク構成を図 2 に示す。NTMobile は DC (Direction Coordinator)、NTM 端末、RS (Relay Server) によって構成される。DC や RS はデュアルスタック

ネットワークに設置し、ネットワークの規模に応じて複数台設置することができる。

- DC (Direction Coordinator)  
DC は仮想 IP アドレスの割り当て管理や、NTM 端末に対してトンネル構築などの指示を出す装置である。NTM 端末に割り当てられる仮想 IP アドレスは一意なアドレスであり、各 DC は自身に割り当てられたアドレス空間から重複が起きないように割り当てを行う [13]。また、DC は Dynamic DNS の機能を包含しており、NTM 端末の A/AAAA レコードに加えて、NTMobile 専用レコード (以下 NTM レコード) を登録することにより、NTM 端末のアドレス情報を管理する。NTM レコードには IPv4 のアドレス情報を記載する NTMv4 レコードと、IPv6 のアドレス情報を記載する NTMv6 レコードが定義されており、それぞれに NTM 端末の FQDN (Fully Qualified Domain Name)、実 IP アドレス、仮想 IP アドレス、NAT の外側の実 IP アドレス、自身のアドレス情報を管理する DC の実 IP アドレスなどが記載されている。
  - NTM 端末  
NTM 端末は移動先のネットワークから割り当てられる実 IP アドレスと、DC から割り当てられる仮想 IP アドレスの 2 種類のアドレスを保持している。仮想 IP アドレスはネットワークに依存しないアドレスであり、NTM 端末が接続先のネットワークを切り替えても変化しない。NTM 端末には常に IPv4 と IPv6 の仮想 IP アドレスが割り当てられており、アプリケーションは仮想 IPv4 アドレスまたは仮想 IPv6 アドレスのどちらかに基づいた通信を行う。これにより、通信中にネットワークを切り替えても、アプリケーションは通信を継続することができる。
  - RS (Relay Server)  
RS は、異なる NAT 配下に位置する NTM 端末同士が通信を行う場合や、NTMobile 非対応端末と通信を行う場合など、特定の状況において通信を中継する装置である。図 2 における NTM 端末 A と移動後の NTM 端末 B のように、通信端末が異なるアドレスファミリのネットワークに接続している場合には、プロトコルの違いから直接通信を行う事ができない。そのため、NTM 端末 A と NTM 端末 B は、デュアルスタックネットワークに設置した RS との間にトンネルを構築し、RS を経由した通信を行う。RS はインターネット上に分散設置することが可能であり、中継負荷や経路の冗長性を考慮してトンネル構築時に最適な RS を選択することができる。
- DC と各端末は信頼関係があることを前提としており、NTMobile で使用されるメッセージ

は各端末間で共有している暗号鍵を用いて暗号化および MAC\*<sup>1</sup>が付加される。また、NTM 端末間や NTM 端末と RS の間で行われるトンネル通信は、トンネル構築時に DC より配布される共通鍵を用いて暗号化および MAC が付加される。

本章では、NTM 端末同士が IPv4 のアプリケーションによる通信を行う際の動作を例に記述する。以後の説明では、通信開始側の NTM 端末を MN (Mobile Node)、通信相手側の NTM 端末を CN (Correspondent Node) とする。また、実 IPv4 アドレスを  $RIP_{4X}$ 、仮想 IPv4 アドレスを  $VIP_{4X}$ 、実 IPv6 アドレスを  $RIP_{6X}$  とし、アドレス情報を管理している DC を  $DC_X$  とする。MN と CN の通信時に用いる Path ID を  $PID_{MN-CN}$ 、暗号化および認証に用いる共通鍵を  $CK_{MN-CN}$  とする。Path ID は NTM 端末間の通信を一意に識別するための識別子である。

### 3.2 アドレス情報の登録

NTM 端末は、端末起動時および接続先ネットワークの切り替え時に、アドレス情報を各自の DC に登録する。MN は FQDN や  $RIP_{4MN}$ 、 $RIP_{6MN}$  など、NTMv4 レコードおよび NTMv6 レコードに登録する情報を記載した Registration Request を  $DC_{MN}$  へ送信する。 $DC_{MN}$  は Registration Request を受信すると、DNS サーバに登録されている MN のリソースレコードを更新し、MN へ応答を返す。IPv4 ネットワークにおいて、Registration Request の IP ヘッダに格納されている送信元 IPv4 アドレスが MN の実 IPv4 アドレスと異なる場合には、MN が NAT 配下に存在すると判断し、送信元 IP アドレスを NAT の外側の実 IPv4 アドレスとして MN の NTMv4 レコードに登録する。また、IPv6 ネットワークでは、インタフェースへ複数の IP アドレスを設定することが可能であり、リンクローカルユニキャストアドレスのように、インターネット上で使用することができない IP アドレスが端末へ割り当てられることがある。この場合には、インターネット上で通信可能なグローバルユニキャストアドレスのみを NTMv6 レコードへ登録する。

登録処理が完了した後、MN と  $DC_{MN}$  は定期的にメッセージを交換することにより、制御メッセージ用の通信経路を確保する (Keep Alive)。これにより、MN が NAT 配下に接続している場合であっても  $DC_{MN}$  は常に MN へ制御メッセージを送信することができる。また、CN についても同様の処理を行い、 $DC_{CN}$  へアドレス情報を登録する。

### 3.3 トンネル構築処理

NTM Mobile では、NTM 端末が通信開始時に行う DNS による名前解決処理や、ハンドオー

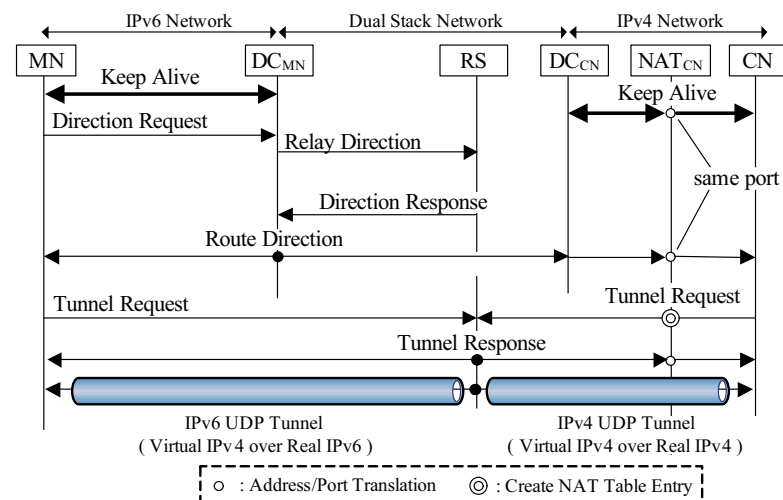


図 3 トンネル構築手順

Fig. 3 Tunnel establishment procedure.

パによる実 IP アドレスの変化を検出した際にトンネル構築処理を実行し、トンネル経路を確立する。MN は DNS による名前解決処理として A レコードの問い合わせを検出した場合には、CN の NTMv4 レコードおよび NTMv6 レコード、AAAA レコードを追加で問い合わせることにより、仮想 IPv4 アドレスなどのアドレス情報を取得する。その後、DNS サーバからの A レコードの応答を一時待避し、取得したアドレス情報をもとにトンネル構築処理を実行する。

図 3 に、IPv6 ネットワークに接続した MN が IPv4 プライベートネットワークに接続した CN との間に通信経路を確立するまでの様子を示す。各端末は以下の手順に従って、MN と CN 間に通信経路を確立する。

- 指示要求 (Direction Request)

MN は  $DC_{MN}$  へ Direction Request を送信し、CN と通信を行うためのトンネルの構築指示を要求する。Direction Request には、MN と CN の実 IP アドレスや仮想 IP アドレスなどの NTMv4/NTMv6 レコードに記載されたアドレス情報が含まれている。 $DC_{MN}$  はこの内容から通信端末の位置関係を認識し、トンネル通信時の経路を決定す

\*1 Message Authentication Code

る。今回は IPv4 ネットワークと IPv6 ネットワーク間の通信であるため、RS を経由した通信経路となる。

- 中継指示 (Relay Direction)

$DC_{MN}$  は、RS に対して MN と CN の通信を中継するよう指示を記載した Relay Direction を送信する。Relay Direction には  $PID_{MN-CN}$  および、MN と CN のアドレス情報が記載されており、これを受信した RS は  $DC_{MN}$  へ応答を返し、MN と CN から送信される Tunnel Request を待機する。

- 経路指示 (Route Direction)

RS からの応答を受信した  $DC_{MN}$  は、MN と CN へ Route Direction を送信し、RS に対して Tunnel Request を送信するよう指示する。Route Direction には  $PID_{MN-CN}$ 、通信相手のアドレス情報、トンネルの構築先の実 IP アドレス、およびトンネル通信時の暗号化に用いる共通鍵  $CK_{MN-CN}$  などが記載されている。

- トンネル構築要求 (Tunnel Request/Response)

MN と CN は  $DC_{MN}$  の指示に従って RS へ Tunnel Request を送信する。NAT 配下に接続した CN から RS へ Tunnel Request を送信することにより、 $NAT_{CN}$  に CN と RS が通信を行うためのマッピング情報が生成され、CN と RS の間に NAT を跨った IPv4 によるトンネルを構築することができる。また、MN が RS へ Tunnel Request を送信することにより、MN と RS 間には IPv6 によるトンネルが構築される。

NTM 端末はカーネル空間に保持しているトンネルテーブルへ、構築したトンネルの Path ID や通信相手の仮想 IP アドレス、トンネルの構築先の実 IP アドレス、暗号化に用いる共通鍵などを登録する。また、RS はカーネル空間にリレーテーブルを保持しており、構築したトンネルの Path ID や転送先の実 IP アドレス、暗号化に用いる共通鍵などを登録する。

以上により、MN と CN が通信を行うための通信経路を確立することができる。A レコードの問い合わせをトリガーに処理を実行していた場合、MN は待避していた DNS サーバの応答に含まれる IP アドレスを  $VIP_{4CN}$  に書き換え、DNS リゾルバへ渡す。これにより、MN のアプリケーションは CN の IPv4 アドレスとして  $VIP_{4CN}$  を認識し、アプリケーション間では仮想 IPv4 アドレスに基づいた通信が開始される。

DNS リゾルバの実装によっては、A レコードの問い合わせが完了した後に AAAA レコードの問い合わせが実行される。この場合にはトンネル構築処理は行わず、構築済みのトンネル情報をもとに CN の仮想 IPv6 アドレスを記載した DNS 応答メッセージを生成し、DNS リゾルバへ渡す。これにより、アプリケーションは CN の IPv6 アドレスとして仮想 IPv6

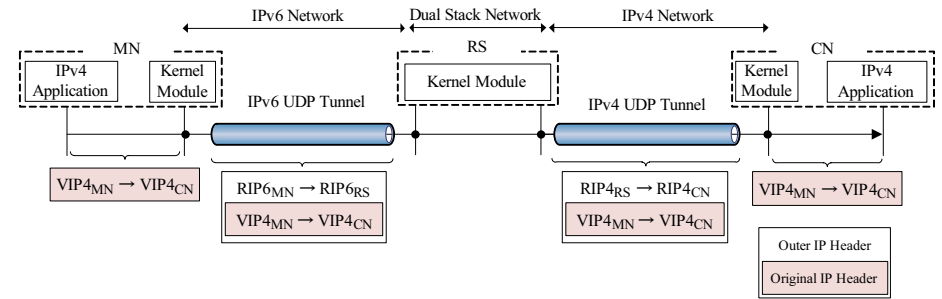


図 4 トンネル通信時のアドレス遷移  
Fig. 4 Address transition in tunneling communication.

アドレスを認識することになる。仮想 IPv6 アドレス宛のパケットが送信された場合には、A レコード問い合わせ時に構築されたトンネルにより送信する。

### 3.4 トンネル通信

IPv6 ネットワークに接続した MN から、IPv4 プライベートネットワークに接続した CN へトンネル通信を行う様子を図 4 に示す。アプリケーションは、名前解決処理により取得した仮想 IPv4 アドレスまたは仮想 IPv6 アドレス宛に通信を開始する。この例ではアプリケーションが仮想 IPv4 アドレスに基づいた通信を行うため、アプリケーションが生成したパケットの送信元アドレスには  $VIP_{4MN}$ 、宛先アドレスには  $VIP_{4CN}$  が記載される。MN は宛先アドレスである  $VIP_{4CN}$  をキーとしてトンネルテーブルを検索し、該当エントリに従ってカプセル化を行う。ここでは、実 IPv6 アドレス  $RIP_{6RS}$  にてカプセル化したパケットを RS へ送信する。なお、カプセル化の際には IP ヘッダと UDP ヘッダ、Path ID などを記載した NTM ヘッダが付加され、共通鍵  $CK_{MN-CN}$  により元パケットすべてが暗号化される。RS は、受信したパケットの NTM ヘッダに記載されている Path ID をキーとしてリレーテーブルを検索し、転送先の経路情報を取得する。RS は該当エントリに従って受信パケットをデカプセル化し、実 IPv4 アドレス  $RIP_{4CN}$  にて再度カプセル化したあと、CN へ送信する。CN はカプセル化されたパケットを受信すると、NTM ヘッダに記載されている Path ID をキーにトンネルテーブルを検索し、該当エントリに従ってデカプセル化および復号処理を行う。その後、抽出したパケットを上位アプリケーションへ渡す。

以上により、IPv6 ネットワークに接続した MN と IPv4 ネットワークに接続した CN 間に通信を行うことができる。アプリケーションは仮想 IPv4 アドレスに基づいた通信を行



うため、NTM 端末が接続しているネットワークの違いによる影響を受けない。通信経路上に NAT が存在している場合であっても、NAT によるアドレス/ポート変換はカプセル化パケットの外側 IP ヘッダと UDP ヘッダに対して行われるため、アプリケーションは NAT に影響されることなく通信を行うことができる。

### 3.5 ネットワーク切り替え時の動作

MN の移動や無線インタフェースの切り替えにより、接続先が異なるネットワークへ切り替わった場合には、3.4 節と同様の手順により CN との間にトンネルを再構築する。トンネル構築時には MN と CN の位置関係に応じて DC が適切なトンネル経路を指示し、エンドツーエンドでトンネルを構築可能な場合には文献 [11, 12] の手順によりトンネルを再構築する。このとき、CN のアドレス情報は取得済みであるため、名前解決処理を省略してトンネル構築のみを実行する。MN と CN のアプリケーションは仮想 IP アドレスに基づいた通信を行っているため、実 IP アドレスが変化してもその影響を受けることなく、通信を継続することができる。また、トンネル経路の切り替えと並行して 3.2 節にて説明した登録処理を行い、DC<sub>MN</sub> に登録したアドレス情報を更新する。常に最新のアドレス情報を DC へ登録することにより、NTM 端末に対する到達性を確保する。

## 4. 実装・性能評価

### 4.1 実装

NTMobile はすでに Android OS を搭載したスマートフォンへ実装しており、IPv4 環境において有効性を確認済みである [18]。本稿では、IPv4 と IPv6 が混在したネットワーク環境における動作検証および性能評価を行うため、各種プログラムを IPv4 と IPv6 の両方に対応するよう拡張した。

NTM 端末はカプセル化を行うカーネルモジュール、ネゴシエーションを行うデーモンプログラム（以後 NTM デーモン）、および仮想インタフェースを実装することにより動作する。カーネルモジュールでは Netfilter によりパケットをフックし、カーネル空間においてカプセル化および暗号化を実行する。NTMobile ではカーネル空間にてカプセル化処理を完結することにより、スループットの低下を抑制している [12]。

NTM デーモンは、DC へのアドレス情報の登録処理やトンネル構築処理を行う。IPv6 ネットワーク移動時における IPv6 アドレス自動生成に要する時間は、ルータから送信される RA (Router Advertisement) の送信間隔による影響が大きいいため、L2 ハンドオーバーを検出した際に RS (Route Solicitation) を送信するよう NTM デーモンへ処理を追加した。

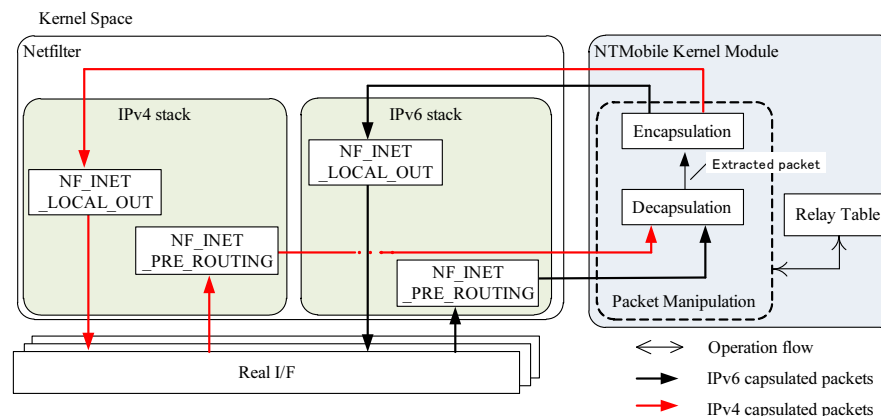


図 5 RS による IPv4/IPv6 ネットワーク間の転送処理  
Fig. 5 Procedure of forwarding between IPv4 and IPv6 networks by RS.

なお、L2 ハンドオーバーおよび IP アドレスの変化は、カーネル空間から送信されるリンク情報およびアドレス情報の変化通知を NTM デーモンが netlink ソケットにて受信することにより検出する。

RS は、トンネル通信の中継を行うカーネルモジュールとネゴシエーションを行うデーモンプログラムを実装することにより動作する。本稿では、新たにカーネルモジュールへ IPv4 ネットワークと IPv6 ネットワーク間のパケット転送機能を実装した。図 5 に RS による IPv4/IPv6 ネットワーク間の転送処理を示す。RS は受信したパケットを Netfilter の NF\_INET\_PRE\_ROUTING にてフックしてカーネルモジュールへ渡すことにより、転送処理を実行する。IPv6 ネットワークから IPv4 ネットワークへの転送を行う際には、受信した IPv6 パケットをデカプセル化し、抽出したパケットを実 IPv4 アドレスにて再度カプセル化したあと、ip\_local\_out 関数により IPv4 の NF\_INET\_LOCAL\_OUT へ渡す。これにより、カプセル化パケットが実インタフェースから IPv4 ネットワークへ送信される。IPv4 ネットワークから IPv6 ネットワークへ転送する際にも、同様の手順により処理する。

### 4.2 実験環境

NTMobile を実装した Android 端末を用いて、IPv4/IPv6 混在環境における接続性の確立およびハンドオーバーの動作検証と性能評価を行った。図 6 と表 1 にネットワーク構成と

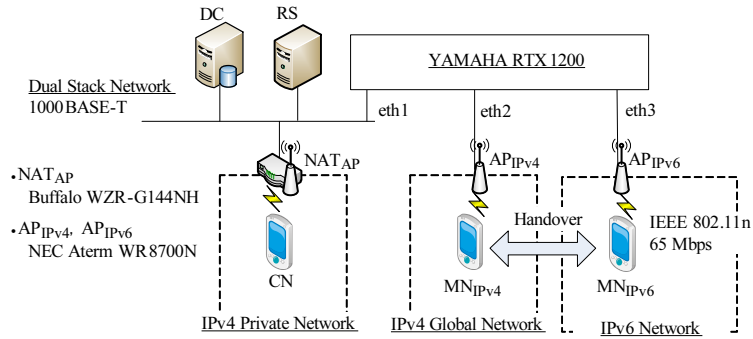


図 6 測定環境  
Fig.6 Evaluation environment.

表 1 装置仕様  
Table 1 Device specifications.

	DC	RS	MN, CN
Hardware	Epson Endeavor NT350	HP s5550j	Sumsung Galaxy S2 ( SC-02C )
OS	Ubuntu 10.04 32bit	Ubuntu 10.04 32bit	Android 2.3.7
Linux Kernel	2.6.32.56	2.6.32.56	2.6.35.7
CPU	Intel Pentium M 1.73GHz	Intel Core i7 2.93GHz	Samsung Exynos Orion Dual-core 1.2GHz
Memory	512MB	10GB	512MB

各装置の仕様を示す。MN と CN は市販の Android スマートフォンであり、NTMobile のカーネルモジュールを実装するために必要な Netfilter や Netlink などの機能を有効にしてカーネルを再構築した。また、通常の Android アプリケーションは Dalvik 仮想マシン上で動作するが、NTM デーモンはネイティブプログラム\*1として実装した。DC および RS は Linux PC により構築した。eth1~3 配下のネットワークはそれぞれデュアルスタックネットワーク、IPv4 ネットワーク、IPv6 ネットワークとして構築し、YAMAHA RTX1200 の VLAN 機能により分割した。各アクセスポイント (AP) は市販のブロードバンドルータであり、AP<sub>NAT</sub> の配下は IPv4 プライベートネットワークとして構築した。AP<sub>IPv4</sub> と AP<sub>IPv6</sub> はルータ機能を無効にして、一般的なアクセスポイントとして動作させた。MN と CN は各 AP へ IEEE 802.11n にて接続し、暗号化および認証機能には WPA/WPA2-PSK (AES)

表 2 端末間の RTT  
Table 2 RTT between the terminals.

	RTT [ms]		
	min	ave	max
MN <sub>IPv4</sub> - DC	1.58	17.66	139.77
MN <sub>IPv4</sub> - CN	4.57	60.63	326.07
MN <sub>IPv6</sub> - DC	1.49	18.01	140.52
MN <sub>IPv6</sub> - RS	2.61	18.54	127.35
CN - DC	5.97	12.28	19.90
CN - RS	4.01	12.29	34.22
DC - RS	0.09	0.21	5.29

を使用した\*2。ネゴシエーション時に使用する暗号化アルゴリズムとして AES-CFB, トンネル通信時には AES-CBC を設定し、認証アルゴリズムは HMAC-MD5, 鍵長 128bit とした。また、測定環境の特性を明確にするために、各端末間の平均 RTT (Round-Trip Time) を測定した。RTT の測定には ping を使用し、1 秒間隔で 64 バイトのパケットを 100 回送受信した。表 2 に測定した端末間の RTT を示す。

今回使用した Galaxy S2 では、DHCP による IPv4 アドレスの取得処理に失敗した際に AP から強制的に切断されてしまい、IPv6 ネットワークへ接続することができなかった。そのため、eth3 配下のネットワークでは、DHCP により IPv4 のリンクローカルアドレス 169.254.0.0/24 を配布するよう設定し、NTM 端末側では 169.254.0.0/24 を 0.0.0.0 として扱うよう、NTM デーモンおよびカーネルモジュールに応急処置を施した。これにより、NTMobile の動作上は eth3 配下のネットワークが IPv6 ネットワークとして扱われる。

### 4.3 ネゴシエーションによるオーバーヘッド

通信開始時に発生するオーバーヘッドを明らかにするために、MN と AP<sub>NAT</sub> へ接続した CN の間に通信経路が確立されるまでに要する時間を測定した。測定は次の 3 ケースにて実施した。

case 1 NTMobile 未実装時、MN を AP<sub>IPv4</sub> へ接続

case 2 NTMobile を実装し、MN を AP<sub>IPv4</sub> へ接続 (IPv4 Grobal to IPv4 Private )

case 3 NTMobile を実装し、MN を AP<sub>IPv6</sub> へ接続 (IPv6 to IPv4)

\*1 CPU が直接解釈可能な形式のバイナリプログラム

\*2 NAT<sub>AP</sub> は IEEE 802.11n ドラフト版準拠。

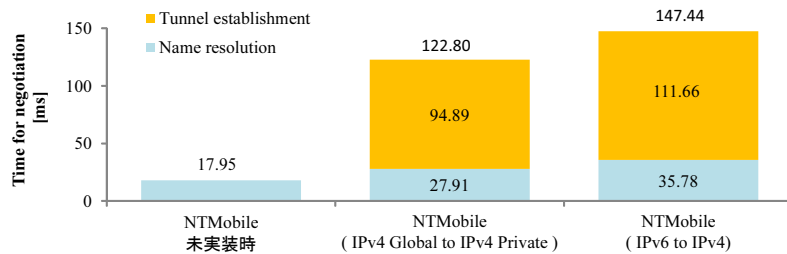


図 7 ネゴシエーションによるオーバーヘッド時間  
Fig. 7 Results of overhead time by negotiation.

測定用に A レコードの問い合わせのみを行う Android 端末向けのネイティブプログラムを作成し、A レコードの問い合わせ前と問い合わせ結果取得後のタイムスタンプの差分から、通信開始時に発生するオーバーヘッドを算出した。なお、NTMobile のネゴシエーション処理に要した時間を測定するために、NTM デーモンにイベントごとのタイムスタンプを出力するようプログラムを追加し、その差分から名前解決処理とトンネル構築処理に要した時間を算出した。

図 7 にネゴシエーションによるオーバーヘッドを 100 回測定した平均値を示す。NTMobile 未実装時の測定結果は通常の名前解決に要する時間であり、17.95 ms で処理が完了した。ただし、この場合は CN に対する接続性を確保することができない。それに対して NTMobile 実装時には、通信開始時に行う名前解決処理に 27.91~35.78 ms を要した。これは A レコードの問い合わせを行った後に、NTMv4 レコード、NTMv6 レコードおよび AAAA レコードを DNS サーバへ問い合わせるためである。

トンネル構築処理に要した時間は、case 2 では 94.89 ms、case 3 では 111.66 ms であった。case 2 では MN と CN との間にエンドツーエンドでトンネルが構築され、case 3 では RS との間にトンネルが構築される。今回の測定環境では、MN と CN のアドレス情報を管理する DC を同一の PC により構築したが、DC<sub>MN</sub> や DC<sub>CN</sub> として異なるネットワークに分散設置することも可能である。その場合には、DC<sub>MN</sub> と DC<sub>CN</sub> 間にて Route Direction の転送が行われるため、トンネル構築処理に要する時間は DC 間の RTT に応じて数十 ms 程度増加すると考えられる。また、NTM 端末が 3G や WiMAX などのネットワークへ接続した際には、NTM 端末と DC や RS 間における RTT が 100 ms 程度増加し、それに

じてネゴシエーションによるオーバーヘッドも増加すると考えられる。

IP を用いて携帯電話のマルチメディアサービスを実現する IMS (IP Multimedia Subsystem) では、端末間におけるセッション制御に SIP (Session Initiation Protocol) を用いている。文献 [19] によると、通信開始時に行うシグナリングに 2~3 秒程度を要するとされている。また、一般的に、Web サイト閲覧時にエンドユーザが許容できる待ち時間は 2 秒程度とされている<sup>\*1</sup>。このような事例から判断すると、NTMobile による通信開始時のオーバーヘッドは許容できる値であり、実用上は問題ないと考えられる。

#### 4.4 通信切断時間の測定

端末のハンドオーバー時には、L2 ハンドオーバーや IP アドレスの取得処理に起因する通信切断時間が発生する。これに伴い、パケットロスが発生してアプリケーション間の通信に影響が及ぶ可能性がある。そこで、MN が CN との通信中に接続先の AP を切り替えることによって発生する通信切断時間を測定した。CN は常に AP<sub>NAT</sub> へ接続しておき、MN の接続先 AP を次のように手動で切り替えた。

case 1 AP<sub>IPv4</sub> 経由での通信中に AP<sub>IPv6</sub> へ切り替える。

case 2 AP<sub>IPv6</sub> 経由での通信中に AP<sub>IPv4</sub> へ切り替える。

MN と CN は iperf<sup>\*2</sup>による TCP 通信を行い、TCP のシーケンス番号の変化および通信パケットから通信切断時間を明らかにした。TCP シーケンス番号の観測には専用のカーネルモジュールを使用し、MN へ実装した。観測用モジュールでは、Netfilter の NF\_INET\_LOCAL\_OUT にてパケットをフックし、TCP シーケンス番号をカーネルメッセージへ出力する。また、パケットの送受信をキャプチャするために、MN と CN へ LAN アナライザアプリケーションである Shark for Root<sup>\*3</sup>をインストールした。なお、キャプチャしたデータの解析には Wireshark<sup>\*4</sup>を使用した。

表 3 にハンドオーバーに伴う通信切断時間を 10 回測定した平均値を示す。case 1 (IPv6 ネットワークへの移動) では AP<sub>IPv6</sub> へ切り替えてから IP アドレスを取得するまでに 1.71 秒を要した。図 8 に AP<sub>IPv6</sub> へ切り替えてからトンネル構築処理が開始されるまでのシーケンスと処理時間の内訳を示す。MN は IPv6 アドレスを生成した後、そのアドレスをターゲットとした NS (Neighbor Solicitation) を送信し、約 1 秒後にトンネル構築処理が開始

\*1 [http://www.akamai.com/html/about/press/releases/2009/press\\_091409.html](http://www.akamai.com/html/about/press/releases/2009/press_091409.html)

\*2 <http://sourceforge.net/projects/iperf/>

\*3 <https://play.google.com/store/apps/details?id=lv.n3o.shark>

\*4 <http://www.wireshark.org/>



表 3 ハンドオーバーに伴う通信中断時間  
Table 3 Results of suspended time by handover.

	Time [sec]	
	case 1 (IPv6 ネットワークへの移動)	case 2 (IPv4 ネットワークへの移動)
L2 ハンドオーバー	0.58	0.47
IP アドレス取得	1.71	0.67
トンネル構築処理	0.17	0.12
TCP の再送待機	1.17	0.66
合計時間	3.63	1.92

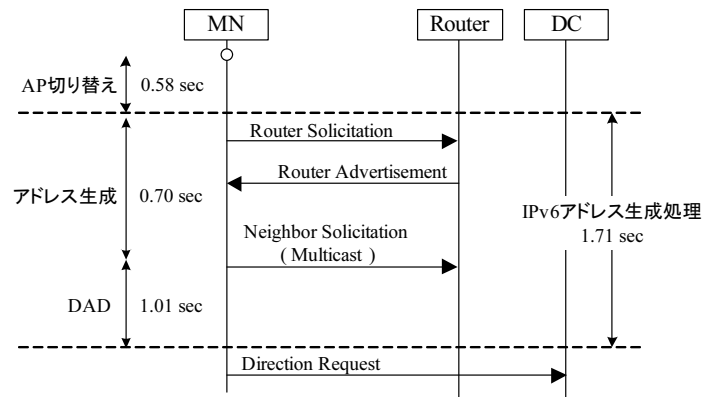


図 8 IPv6 におけるハンドオーバーの処理  
Fig. 8 Handover flow in IPv6.

された。これは、アドレス重複チェック DAD (Duplicated Address Detection) によるものである [20]。MN にて生成された IPv6 アドレスは、DAD による処理が完了するまでの一定間は使用することができない。この処理が完了した後、NTMobile によるトンネル構築処理が開始されたが、全体の 5% 程度の時間で処理を完了した。

case 2 (IPv4 ネットワークへの移動) では AP<sub>IPv4</sub> へ切り替えてから 1.92 秒間、通信が切断了。図 9 に IPv4 ネットワークへのハンドオーバーによる TCP シーケンス番号の変化の様子を示す。AP を切り替えてから IP アドレスを取得するまでに、TCP による再送が 2 回行われた。MN の接続先を AP<sub>IPv6</sub> から AP<sub>IPv4</sub> へ切り替えた後、L2 ハンドオーバーお

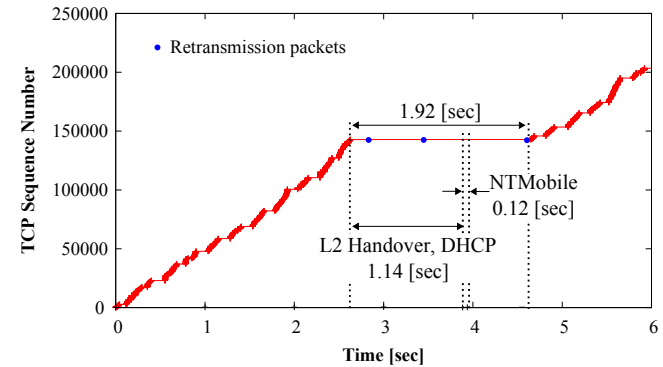


図 9 IPv4 ネットワークへのハンドオーバーによる TCP シーケンス番号の変化  
Fig. 9 Changes of TCP sequence number by handover.

よび DHCP による IP アドレス取得処理が行われ、1.14 秒を要した。また、IP アドレス取得後に NTMobile によるトンネル再構築処理が実行され、0.12 秒を要した。トンネルの再構築完了後、すぐにはアプリケーションによる通信は再開されず、0.66 秒後にアプリケーションによる通信が再開された。これは、TCP の再送制御が機能したものと考えられる。

文献 [21] によると、DHCP にて IP アドレスを取得する際には、ネットワークに接続してから 1~10 秒程度待機したあとに処理を開始すべきであるとされている。今回使用した Galaxy S2 では AP へ接続してから 0.3 秒後に DHCP による処理が開始されたが、この待機時間は DHCP クライアントに依存するため、使用する端末によっては IP アドレスの取得時間が数倍に増加し、それによって通信切断時間が増加する可能性がある\*1。

測定結果より、ハンドオーバー時に行う NTMobile による処理はわずかな時間で完了するが、L2 ハンドオーバーと IP アドレスの取得処理に 1.14~2.29 秒を要し、通信切断時間の半分以上を占めていることがわかった。また、DHCP クライアントの実装によっては、IP アドレスの取得処理に要する時間がさらに増加することが予想される。IEEE 802.21 [22] ではリンク情報を抽象化することにより、異なる無線システム間においてシームレスハンドオーバーを実現することができる。文献 [23] では、SIP-based Mobility と IEEE 802.21 を組み合わせることにより、ハンドオーバー時に発生する通信切断時間を 4 秒から 20 ms まで短縮している。また、文献 [24] では、DHCP 処理中に使用可能な一時アドレスを導入する

\*1 Motorola Xoom では、AP への接続を完了してから 1~5 秒後に DHCP による処理が開始された。

ことにより、IPアドレスの取得に要する時間を5分の1程度に削減している。IETFではDAD処理を最適化するOptimistic DADが定義されており、生成直後のアドレスの状態をOptimisticとすることにより、生成したIPv6アドレスを即座に使用可能にする[25]。これにより、DADによる待機時間をなくすることができるが、端末およびルータがOptimistic DADに対応していなければならない。一方、ルータに依存しないDAD回避方式が提案されており、文献[26]ではアドレス生成に起因する通信切断時間を20分の1に削減している。

今後は、このような技術を用いることにより通信切断時間を削減するための検討が必要である。また、Wi-Fiや3Gなどの異なる無線インタフェースを使用し、ハンドオーバー前にネゴシエーションを完了することにより、通信切断時間やパケットロスをなくすることができると考えられる[27]。

## 5. おわりに

本稿では、IPv4/IPv6混在環境において相互接続と移動透過性を実現するNTMobileをAndroid端末へ実装し、動作検証および性能評価を行った。動作検証により、NATやIPv4/IPv6ネットワークの違いに影響されることなく、接続性の確保と移動透過性を実現可能なことを確認した。また、NTMobile実装時には通信開始時に一定のオーバーヘッドが発生するものの、実用上は問題ないことを示した。NTMobileはNAT越え問題やIPv4とIPv6の混在といった現在のIPネットワークにおける二つの課題を解決し、確実に接続性を確保することができる。ハンドオーバー時にはL2ハンドオーバーやIPアドレス取得処理に起因する通信切断時間が発生するため、通信切断時間が発生しないシームレスなハンドオーバーを実現する手法の検討が必要である。また、今後は3GやWiMAXなどの実環境において、通信性能およびスケーラビリティの評価を行っていく予定である。

## 参考文献

- 1) Rosenberg, J.: Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols, *RFC 5245, IETF* (2010).
- 2) 鈴木秀和, 宇佐見庄五, 渡邊 晃: 外部動的マッピングによりNAT越え通信を実現するNAT-fの提案と実装, 情報処理学会論文誌, Vol.48, No.12, pp.3949–3961 (2007).
- 3) 宮崎 悠, 鈴木秀和, 渡邊 晃: 端末の改造が不要なNAT越え通信システムNTSSの提案と評価, 情報処理学会論文誌, Vol.51, No.9, pp.1234–1241 (2010).
- 4) Le, D., Fu, X. and Hogrefe, D.: A Review of Mobility Support Paradigms for the

- Internet, *IEEE Communications Surveys*, Vol.8, No.1, pp.38–51 (2006).
- 5) C.Perkins, E.: IP Mobility Support for IPv4, *RFC 3220, IETF* (2002).
- 6) Johnson, D., Perkins, C. and Arkko, J.: Mobility Support in IPv6, *RFC 3775, IETF* (2004).
- 7) 相原玲二, 藤田貴大, 前田香織, 野村嘉大: アドレス変換方式による移動透過インターネットアーキテクチャ, 情報処理学会論文誌, Vol.43, No.12, pp.3899–3897 (2002).
- 8) 関 顕生, 岩田裕貴, 森廣勇人, 前田香織, 近堂徹, 岸場清悟, 西村浩二, 相原玲二: IPv4 拡張した移動透過通信アーキテクチャMAT の設計と性能評価 IPv4 拡張した移動透過アーキテクチャ MAT の設計と実装, 情報処理学会論文誌, Vol.52, No.3, pp.1323–1333 (2011).
- 9) 竹内元規, 鈴木秀和, 渡邊 晃: エンドエンドで移動透過性を実現する Mobile PPC の提案と実装, 情報処理学会論文誌, Vol.47, No.12, pp.3244–3257 (2006).
- 10) Schulzrinne, H. and Wedlund, E.: Application-layer mobility using SIP, *IEEE Globecom '00 Workshop. 2000 IEEE Service Portability and Virtual Customer Environments (IEEE Cat. No.00EX498)*, Vol.1, No.2, pp.29–36.
- 11) 鈴木秀和, 水谷智大, 西尾拓也, 内藤克浩, 渡邊 晃: NTMobileにおける相互接続性の確立手法と実装, マルチメディア, 分散, 協調とモバイル (DICOMO2011) シンポジウム論文集, Vol.2011, No.1, pp.1339–1348 (2011).
- 12) 内藤克浩, 西尾拓也, 水谷智大, 鈴木秀和, 渡邊 晃, 森香津夫, 小林英雄: NTMobileにおける移動透過性の実現と実装, DICOMO2011 論文集, Vol.2011, No.1, pp.1349–1359 (2011).
- 13) 西尾拓也, 内藤克浩, 水谷智大, 鈴木秀和, 渡邊 晃, 森香津夫, 小林英雄: NTMobileにおける端末アドレスの移動管理と実装, マルチメディア, 分散, 協調とモバイル (DICOMO2011) シンポジウム論文集, Vol.2011, No.1, pp.1139–1145 (2011).
- 14) 上醉尾一真, 鈴木秀和, 内藤克浩, 渡邊 晃: IPv6 ネットワークにおけるNTMobileの検討, 情報処理学会研究報告, Vol.2011-MBL-59, No.9, pp.1–7 (2011).
- 15) Soliman, H.: Mobile IPv6 Support for Dual Stack Hosts and Routers, *RFC 5555, IETF* (2009).
- 16) Levkowitz, H. and Vaarala, S.: Mobile IP Traversal of Network Address Translation (NAT) Devices, *RFC3519, IETF* (2003).
- 17) Wakikawa, R., Kuntz, R., Zhu, Z. and Zhang, L.: Global HA to HA Protocol Specification, *draft-wakikawa-mext-global-haha-spec-02, IETF* (2011).
- 18) 上醉尾一真, 鈴木秀和, 内藤克浩, 渡邊 晃: NTMobileのAndroid端末への実装と評価, 情報処理学会研究報告, Vol.2012-MBL-62, No.19, pp.1–8 (2012).
- 19) Munir, A., Member, S. and Gordon-ross, A.: SIP-Based IMS Signaling Analysis for WiMax-3G Interworking Architectures, *IEEE TRANSACTIONS ON MOBILE COMPUTING*, Vol.9, No.5, pp.733–750 (2010).
- 20) Thomson, S. and Narten, T.: IPv6 Stateless Address Autoconfiguration,

*RFC2462,IETF* (1998).

- 21) Droms, R.: Dynamic Host Configuration Protocol, *RFC3519 ,IETF* (2003).
- 22) IEEE 802.21: MEDIA INDEPENDENT HANDOVER SERVICES,  
<http://www.ieee802.org/21/>.
- 23) Dutta, A., Das, S., Famolari, D., Ohba, Y., Taniuchi, K., Kodama, T. and Schulzrinne, H.: Seamless Handover across Heterogeneous Networks - An IEEE 802.21 Centric Approach, *WPMC2005*, Vol.31, No. TM11-4, pp.VI-31—VI-35 (2005).
- 24) Forte, A.G., Shin, S. and Schulzrinne, H.: Improving Layer 3 Handoff Delay in IEEE 802 . 11 Wireless Networks, *WICON '06 Proceedings of the 2nd annual international workshop on Wireless internet*, No.12, pp.1-8 (2006).
- 25) Moore, N.: Optimistic Duplicate Address Detection (DAD) for IPv6, *RFC4429,IETF* (2006).
- 26) 寺澤圭史, 鈴木秀和, 渡邊 晃 : IPv6 における Mobile PPC の実現と評価, 情報処理学会研究報告, Vol.2010-MBL-52, No.5, pp.1-8 (2010).
- 27) 福山陽祐, 鈴木秀和, 渡邊 晃 : IPv4 移動体通信において携帯電話網と無線 LAN 間をシームレスに移動する方式の提案, マルチメディア, 分散, 協調とモバイル (DI-COMO2011) シンポジウム論文集, Vol.2011, No.1, pp.1115-1120 (2011).